REPORT DOCUMENTATION PAGE

Form Approved
OBM No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
June 1998

3. REPORT TYPE AND DATES COVERED
Proceedings

4. TITLE AND SUBTITLE
Performance of Barotropic Ocean Models on Shared and Distributed Memory Computers

5. FUNDING NUMBERS
Job Order No.
Program Element No. 0602435N
Project No.
Task No.
Accession No.

6. AUTHOR(S)
S. Piacsek, A. Wallcraft, [1]P. Jayakumar, [2]L. Lonergan, and [3]M. Young

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Research Laboratory
Oceanography Division
Stennis Space Center, MS 39529-5004

8. PERFORMING ORGANIZATION REPORT NUMBER
NRL/PP/7322--98-0022

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

19991115 110

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Second International Workshop on Software Engineering and Code Design in Parallel Meteorological and Oceanographic Applications, 15-18 June 1998, Scottsdale, AZ
[1]University of Southern Mississippi, Stennis Space Center, MS 39529-5004, [2]Northrup Grumman Corporation, Stennis Space Center, MS 39529-5004, [3]Naval Research Laboratory, Code 5593, Washington, DC 20375

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The efficiency of explicit time integration schemes for barotropic models of the Mediterranean were investigated, in context of the vectorization and parallel modeling approaches employed on different architectures. The main focus of interest was the scalability and MFlops output of the codes as a function of domain size.

For simulations with real winds, mesh sizes ranged from 25 km down to 1.8 km (grids of 180x64 to 2048x1024), with the coarse resolution resolving only major straits like that of Sicily, and the high resolution even narrow straits like Gibraltar and Messina. Since the memory requirement of these grids only ranged to 70 Mbytes, we also performed simulations with idealized, precomputed winds for which mesh sizes ranged down to 280m, to produce a total memory requirement of 4 Gbytes. The analysis and interpretation of the latter results for the Mediterranean has not been performed yet. The explicit scheme consisted of the leap-frog scheme for the Coriolis, pessure gradient and advection terms, and 'lagged' times for the diffusion terms. The platforms utilized included the CM500-E (with CMF), the Cray C90 and T90 (with FT90 -O3 auto tasking), the Cray T3E (with HPF and MPI), the SGI Origin2000 (with f77 -pfa -O2 power fortran, HPF and MPI), the IBM SP2 (with HPF) and the Sun Global Works (with HPF).

The MPI version of this code employed a 2-D tiling decomposition, and parallel runs were performed up to 512 processors on the T3E and up to 64 processors on the SGI Origin. The T3E 512 processors achieved an 82% scaling efficiency of 100% vs. 32 processors, but less than linear for smaller number of processors. The auto-tasking versions were quite efficient even for small program sizes (17 Mb) and for small number of processors, with the SGI -pfa compiler option (with -O2 optimization) giving scalings of 1.9, 3.7, and 15.4 for 2, 4 and 16 CPU's, respectively, while the Cray T90 -O3 option (with FT90) gave scalings of 3.6 and 6.6 for 4 and 8 processors, respectively.

14. SUBJECT TERMS
barotropic models, Mediterranean, Cray C90, Cray T90, Cray T3E, IBM Sp2, parallel machines, and numerical schemes

15. NUMBER OF PAGES
17

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

Piacsek et al

GSFC/CP—1998–206860

Second International Workshop on Software Engineering and Code Design in Parallel Meteorological and Oceanographic Applications

Matthew O'Keefe, University of Minnesota
Christopher Kerr, International Business Machines

Proceedings of a workshop sponsored by the U.S. Department of Energy, Office of Biological and Environmental Research; the Department of Defense, High Performance Computing and Modernization Office; and the NASA Goddard Space Flight Center, Seasonal-to-Interannual Prediction Project, and held at the Camelback Inn, Scottsdale, Arizona June 15–18, 1998

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

June 1998

PERFORMANCE OF BAROTROPIC OCEAN MODELS ON SHARED AND DISTRIBUTED MEMORY COMPUTERS

S. Piacsek and A. Wallcraft
Naval Research Laboratory, Code 7320
Stennis Space Center, MS 39529-5004, USA
piacsek@nrlssc.navy.mil, +1 228-688-5316, FAX: +1 228 688-4759
P. Jayakumar
University of Southern Mississippi/NAVO-MSRC PET Program
Stennis Space Center, MS 39529-5004, USA
L. Lonergan
Northrup Grumman Corporation/NAVO-MSRC PET Program
Stennis Space Center, MS 39529-5004, USA
M. Young
Naval Research Laboratory, Code 5593
Washington, DC 20375, USA

Abstract

The efficiency of explicit time integration schemes for barotropic models of the Mediterranean were investigated, in context of the vectorization and parallel modeling approaches employed on different architectures. The main focus of interest was the scalability and MFlops output of the codes as a function of domain size.

For simulations with real winds, mesh sizes ranged from 25 km down to 1.8 km (grids of 180x64 to 2048x1024), with the coarse resolution resolving only major straits like that of Sicily, and the high resolution even narrow straits like Gibraltar and Messina. Since the memory requirement of these grids only ranged to 70 Mbytes, we also performed simulations with idealized, precomputed winds for which mesh sizes ranged down to 280m, to produce a total memory requirement of 4 Gbytes. The analysis and interpretation of the latter results for the Mediterranean has not been performed yet. The explicit scheme consisted of the leap-frog scheme for the Coriolis, pressure gradient and advection terms, and 'lagged' times for the diffusion terms. The platforms utilized included the CM500-E (with CMF), the Cray C90 and T90 (with FT90 -O3 auto-tasking), the Cray T3E (with HPF and MPI), the SGI Origin2000 (with f77 -pfa -O2 power fortran,HPF and MPI), the IBM SP2 (with HPF) and the Sun Global Works (with HPF).

The MPI version of this code employed a 2-D tiling decomposition, and parallel runs were performed up to 512 processors on the T3E and up to 64 processors on the SGI Origin. The T3E 512 processors achieved an 82 % scaling efficiency relative to 32 CPU's. The SGI 64 processors achieved a scaling efficiency of 100 % vs. 32 processors, but less than linear for smaller number of processors. The auto-tasking versions were quite efficient even for small program sizes (17 Mb) and for small number of processors, with the SGI -pfa compiler option (with -O2 optimization) giving scalings of 1.9, 3.7, and 15.4 for

2, 4 and 16 CPU's, respectively, while the Cray T90 -O3 option (with FT90) gave scalings of 3.6 and 6.6 for 4 and 8 processors, respectively.

MFlops output reached 11.7 GFlops for the 512 node T3E, and 7.4 GFlops for the 128 node O2K.

1 Introduction

The recent advances in high-performance computing, especially on massively- parallel machines, have encompassed ocean models as well. These advances also include the reformulation and design of numerical schemes especially suited for parallel machines. On the one hand, progress has been achieved by porting older codes to new architectures; on the other hand, some codes have been designed from scratch for the new machines. In the 1992-1998 period at the Naval Research Laboratory, great advances were made in ocean modeling on a series of parallel computers. The principal machine used up-to-now by the group was the CM5-E, with 256 nodes and a total memory of 4 Gigawords; currently they are the Cray T3E, with 512 nodes and a total memory of 128 GB, and the SGI Origin2000, with 128 processors and 32 GB of memory.

The large memory and throughput of these machines enabled us to push the frontiers of ocean modeling much further, solving some problems important for Navy environmental prediction and climate simulation. Among these were the simulation of the reapparance of the 1983 El Nino 8 years later in the western Pacific, having traced the westward propagation of the constituent Rossby waves with sufficient spatial resolution and phase accuracy [Jacobs et al,1994]. The crucial factor in ocean modeling has always been resolution, both in the horizontal and vertical. High resolution reduces truncation and dispersion errors; in addition, it allows the size of the friction coefficient to be kept small, increasing the amplitudes of the ocean currents. A further advantage is the better resolution of gradients and the decrease in eddy sizes that can be resolved, extending the solution spectrum and energy cascade, crucial for long term simulations. A large part of the lateral friction employed in ocean models is necessary for numerical rather than physical reasons, to keep solutions stable and smooth.

There appeared to be two problems in the beginning of our parallel computations that had to be tackled: (a) how to migrate existing ocean models from a shared memory to a distributed memory computer; (b) a rethinking of the numerical techniques and algorithmic approaches to take advantage of the massively parallel nature of the new computers. On data-parallel machines, such as the CM5, this migration required a complete rewrite of the existing model codes into CM Fortran (CMF). On distributed memory machines, the use of HPF (High Performance Fortran) was facilitated by the existing CMF codes, from which HPF codes could be generated in about a week or two. Some of the large 3-D codes were ported directly from the Cray C90 to the T3D and T3E via message-passing: this typically involved inserting MPI message-passing function calls into regular f77 subroutines. On massively parallel machines, such things as communicating between neighboring grid points are major

issues, and the problem revolves around a trade-off between keeping all processors busy ("load balancing") versus keeping communication down between processors as much as possible ("locality management"). Furthermore, certain numerical schemes that rely on recursion relations (e.g. the well-known tridiagonal algorithm) encounter the same problems of conflict and inefficiency on parallel machines as they do on vector machines. Thus alternate formulations for inverting tridiagonal matrices (e.g. the Buzbee-Golub cyclic reduction technique [Buzbee et al,1971; Schwarztrauber,1977]) had to be invoked.

For the major part of the ocean modeling subroutines, it was relatively easy to rewrite the existing codes; however, certain Helmholtz solvers required major recoding efforts to arrive at an efficient program representation (Wallcraft and Moore, 1997). One problem we found with parallel computers is that performance on certain codes was degraded by necessary actions that are required at only a small number of grid points (e.g. the computation of boundary conditions), since all other processors are idle while these actions are performed.

For a review of some recent efforts in parallelizing ocean models, the reader is referred to Smith et al (1992), Dukowicz et al (1993), Piacsek and Wallcraft (1993), Bleck et al (1995), Webb (1995), Oberhuber and Ketelsen (1995), Wallcraft and Moore (1997) and Ma et al (1998).

The present report will focus on barotropic ocean models that employ explicit time integration, and do not require the use of a Helmholtz solver. In Section 1 we introduce barotropic models and give will a rationale for their use. In Section 2 we will give model details, including the numerical scheme for the explicit time stepping version, and some sample results in the Mediterranean. In Section 3 we present some performance figures on various parallel platforms.

2 Barotropic Models

2.1 Utility of Barotropic Models

Barotropic ocean models are 2-D and represent the ocean with one deformable layer, obtained upon integrating vertically the 3-D equations of a hydrostatic ocean. They include topography of the ocean bottom and (generally) a uniform density. All 3-D ocean models contain a barotropic mode (i.e. the vertically averaged motion). For a discussion of these topics, the readers are referred to Bryan,1969,1979; Madala and Piacsek,1977; Blumberg and Mellor,1987; Wallcraft,1991 and Dukowicz and Smith,1994.

It may be asked why barotropic modelling is done at all except in conjunction with baroclinic modelling? We can give three reasons immediately:

(a) in certain oceanic regions, especially in the winter season, deep convective mixing can occur which will tend to homogenize the density field over most of the depth range, so that the barotropic part may represent a significant if not the dominant component of the total circulation. Thus we can gain a useful insight into its patterns by using only a barotropic model, at a much lower computational cost.

(b) tidal forces create the greatest response in the barotropic mode, and their response must be modeled in shallow waters.

(c) The third, and probably most important, point is that the free surface elevation couples directly to the barotropic mode. The associated surface gravity waves, with their fast propagation velocities, can cause great problems for the numerical computations. For various numerical and computational reasons, this makes the solution of the barotropic equations the most CPU intensive, and hence expensive. Hence it becomes very cost-effective to study the efficiency and accuracy of the numerical techniques on parallel platforms in the 2-D setting of barotropic models, rather than in a costly 3-D baroclinic setting.

(d) The central role of the free surface in barotropic models becomes even more important when altimeter measurements of the sea surface elevation are used as part of the initialization/updating procedure for real-time prediction. This is because the information about the free surface elavation is first passed through the barotropic mode before its pressure effects are felt by the whole water column. Thus barotropic models are used to estimate the atmospheric-pressure induced sea surface elevations, not only the simple inverse barometer effect, but the so-called 'non-isostatic' response due to moving weather patterns. Both this 'non-isostatic' response, and the surface elevations due to tidal forcing, are needed to calibrate altimeter measurements of sea surface height [Kantha, 1995].

2.2 Model Equations for a Barotropic Ocean

To shorten and simplify the numerical description, we will present only the finite difference form of the relevant equations in Cartesian coordinates, using constant horizontal friction coefficients; extension to spherical geometry and variable friction coefficients is straightforward. We further assume a constant density, and that the ocean depth is much greater than the free surface elevation, so that only a linear form of the continuity equation needs to be utilized. In the same vein, because of the smallnes of barotropic currents in deep water, we omit the nonlinear advection terms in this description, though they were included in the code and the simulations.

We will use the nonlinear bottom friction formulation (6), and locate the variables on a staggered mesh called the Arakawa C-grid [Wallcaft,1991]. In this arrangement the pressure p and height h variables are located at the center of the mesh boxes, and the mass tansports U and V at the center of the box boundaries facing the x and y directions, respectively.

$$\begin{aligned}\frac{U_{ij}^{n+1}-U_{ij}^{n-1}}{\Delta t} &= fV_{ij}^{n}-\frac{gH}{\Delta x}(h_{i+1/2j}-h_{i-1/2j})^{n} \\ &+\frac{A}{\Delta x^{2}}(U_{i+1j}+U_{i-1j}+U_{ij+1}+U_{ij-1}-4U_{ij})^{n-1} \\ &+(\tau_x)_{ij}-C_d.U_{ij}^{n-1} \qquad (1)\end{aligned}$$

$$\frac{V_{ij}^{n+1} - V_{ij}^{n-1}}{\Delta t} = -fU_{ij}^{n} - \frac{gH}{\Delta y}(h_{ij+1/2} - h_{ij-1/2})^{n}$$
$$+\frac{A}{\Delta y^2}(V_{i+1j} + V_{i-1j} + V_{ij+1} + V_{ij-1} - 4V_{i,j})^{n-1}$$
$$+(\tau_y)_{ij} - C_d.V_{ij}^{n-1} \qquad (2)$$

$$\frac{h_{ij}^{n+1} - h_{ij}^{n-1}}{\Delta t} = -\frac{(U_{i+1/2j} - U_{i-1/2j})^{n}}{\Delta x} - \frac{(V_{ij+1/2} - V_{ij-1/2})^{n}}{\Delta y} \qquad (3)$$

where

$\vec{U} = (U, V)$ - mass transports in the x- and y-directions, respectively

$\vec{\tau_w} = [(\tau_w)_x, (\tau_w)_y]$ - wind stress components

$\vec{\tau_b} = [(\tau_b)_x, (\tau_b)_y]$ - bottom stress components

h - free surface elevation

$f = 2\Omega \sin\theta$ - Coriolis parameter for latitude θ

$H(x, y)$ - topography (ocean depth)

A - coefficient of lateral friction

The bottom stress τ_b can be related in a linear or nonlinear way to the bottom velocity, which in this case has to be replaced by the depth-mean averaged velocity.

The use of a lateral friction coefficient is a general requirement for modeling all hydrodynamic processes that have strong nonlinearities, and as such it becomes a necessary part of ocean models as well. Such friction is also necessary for physical reasons, to represent the subgrid-scale mixing processes.

We assume closed boundaries for our rectangular domain, for which the relevant conditions are U = V = 0. Note that the vanishing of the depth H on land precludes having to specify the gradients of h, and no loss of parallel efficiency will occur.

2.3 Explicit Time Integration

Explicit time integration schemes are attractive because they do not involve matrix inversion or the use of iterative solvers. Though we avoid having to use a matrix inverter to solve for the values at t^{n+1}, we pay the price by being restricted in the time step we can take. The size of the time step one can march with is governed by the well-known Courant-Friedrichs-Levy (CFL) stability condition. For wave equations the time step is limited by the wave speed, in this case the speed of the surface gravity waves c_w, and is given by

$$\Delta t < \Delta x / c_w \qquad (4)$$

Typically, $c_w = \sqrt[2]{gH}$ of the gravity waves exceeds 200m/sec in basins of depth $> 4000m$, so Δt is of order 60 sec for a spatial resolution of 14 km normally associated with eddy-resolving basin-scale (1/8 deg) ocean models.

We must also give expressions for the bottom stress τ_b in terms of the velocity components. In the non-linear approach, the bottom stress takes the form

$$\tau_{bx} = C_d \cdot |U|U, \quad \tau_{by} = C_d \cdot |U|V \tag{5}$$

with the value of the drag coefficient C_d taken to be either .0025 or .0050, depending on the author.

2.4 Barotropic Results in the Mediterranean

We have found that for this simple 2-D explicit code there were no impediments to either vectorization or parallelization. Our first parallel experiments were on the CM5. After the basic conversion from f77 to CMF, including calls to MAXVAL, SUM, etc., an attempt was made to speed up the code by studying serializing or parallelizing the different spatial dimensions, the size of these dimensions, and the handling of the boundary conditions. These have been reported on in Piacsek and Wallcraft (1993).

Using the CM5 code, we carried out simulations of the wind-driven circulation in the Mediterranean, including the non-isostatic response to moving atmospheric pressure gradients. For simulations with real winds on all platforms, mesh sizes ranged from 25 km down to 1.8 km (grids of 180x64 to 2048x1024), with the coarse resolution resolving only major straits like that of Sicily, and the high resolution even narrow straits like Gibraltar and Messina. Since the parallel versions of the wind interpolation routines have not yet been installed, and 6-hourly forcing at the very high resolution presented a forbidding amount of data transfer and storage, we ran the higher resolution cases only with analytic winds.

The model grid for the experiments presented here was 1024x512, giving a horizontal grid size of 3.5 km. The experiments were carried out on the 256-cpu CM500-E at NRL-DC, as well as on the various partitions of the CM5 at Minnesota (it ran even on the 64-cpu partitions). The horizontal diffusivity A was taken to be 50 m^2/sec. The model was forced with GCM-derived synoptic winds obtained from central weather prediction sites, and run typically for 60 days to equilibrium.

Figure 1 shows the transport vectors for the barotropic circulation in the Tyrrhenian Sea for the month of November 1994, The development of a strong cyclonic gyre in the southern half of the basin as winter approaches is quite evident.

Figure 2 shows the transport vectors for the barotropic circulation in the NE corner of the Eastern Mediterranean Basin for the four seasons of 1994. The strong effect of the topography, the so-called 'topographic steering' of the barotropic currents, is quite evident. The well-known, intense cyclonic gyre near the island of Rhodos (the 'Rhodes gyre') is well represented with only the barotropic mode, as is the westward

moving Anatolian current south of Turkey. The main seasonal changes appear to be the appearance of an anticyclonic gyre east of the Rhodes Gyre, and the tendency of the Rhodes Gyre to become an asymmetric bi-polar vortex pair, in the winter months.

3 MPI Version of Barotropic Code

The MPI version of this code employed a 2-D tiling decomposition, and parallel runs were performed up to 512 processors on the T3E and up to 64 processors on the SGI Origin. Since the memory requirement of the grids for the GCM wind simulations ranged to 70 Mbytes, we also performed simulations with idealized, precomputed winds for which mesh sizes ranged down to 280m, to produce a total memory requirement of 4 Gbytes. The analysis and interpretation of the latter results for the Mediterranean has not been performed yet.

In our initial phase of code development, we have used blocking send and receive calls. All processors were sending messages in parallel, but there was no overlap with any computations. The code sections shown below are not complete, citing only the statements relevant to illustrating the MPI approach. In the same vein, the number of processors and mesh dimensions are only illustration.

3.1 MPI Initialization

```
      PROGRAM BTPROGRAM_MPI
      include "mpif.h"
C
      parameter(nprocx = 8,nprocy=8)
      PARAMETER  (NX=2049,NY=1025)
      parameter(MyNX=((NX-2)/nprocx)+2)
      parameter(MyNY=((NY-2)/nprocy)+2)
C
      common /n1/ comm_2d, coords, left, right, above, down
      common /n2/ strided
C
      integer rank, size, ierr
      integer comm_2d, coords(2), left, right, above, down
      integer strided
      logical shift_up, shift_down, shift_left, shift_right
C
```

3.2 The Forecast Routine for the X-Transport U

```
      SUBROUTINE FCST_UVH
C
      integer left,right,above,down, coords(2),comm_2d,strided
      integer nprocx, nprocy,IS,JS
```

```
      logical shift_up, shift_down, shift_left, shift_right
      common /n1/ comm_2d, coords, left, right, above, down
      common /n2/ strided

      shift_up = .true.
      shift_down = .true.
      shift_left = .true.
      shift_right = .true.

            JS = 2
      if(coords(2) .eq. 0)  JS = 1
            IS = 2
      if(coords(1) .eq. 0)  IS = 1
      DO 100 J = JS,JJ1
      DO 100 I = IS,II1
      UU(I,J)   = U(I,J)
      VV(I,J)   = V(I,J)
      UVEL(I,J) = ZU(I,J)/HT(I,J)
      VVEL(I,J) = ZV(I,J)/HT(I,J)
  100 CONTINUE
C
      call shift_data(ZH,mynx,myny,
     & .false., .false., shift_left, .false.)
      call shift_data(UU,mynx,myny,
     &shift_up, shift_down, shift_left, shift_right)
      call shift_data(UVEL,mynx,myny,
     & shift_up, shift_down, shift_left, shift_right)
       call shift_data(ZU,mynx,myny,
     & .false., .false.,shift_left,shift_right)
       call shift_data(VP,mynx,myny,
     & shift_up, .false., .false., .false.)
      IL = II1
      if(coords(1) .eq. nprocx-1)  IL  = II2
C
      DO 200  J = 2,JJ1
      DO 200  I = 2,IL
      IF (MASKU(I,J).NE. 0)THEN
      U(I,J) = U(I,J)
     1     + A2(I,J) *ZV(I,J)
C    1     + A24(I,J)*(ZV(I+1,J)+ZV(I+1,J-1)+ZV(I,J+1)+ZV(I,J))
     2     - A31(I,J)*(ZH(I+1,J) - ZH(I,J))
C    2     + A33(I,J)*(HA(I+1,J) - HA(I,J))
     3     + A41     *(UU(I+1,J) + UU(I-1,J) - 2.*UU(I,J))
     4     + A42     *(UU(I,J+1) + UU(I,J-1) - 2.*UU(I,J))
     5     + A5*(TAUX(I,J) - TAUBX(I,J)) - A6*UU(I,J)
```

```
     6     - A71*((ZU(I+1,J)+ZU(I,J))*UVEL(I+1,J)
     7          - (ZU(I,J)+ZU(I-1,J))*UVEL(I-1,J))
     8     - A72* (VP(I,J)         *UVEL(I,J+1)
     9           - VP(I,J-1)       *UVEL(I,J-1))
        ENDIF
  200 CONTINUE
C
```

The data shift routines are then called from the subroutine detailed in the next section.

3.3 The Data Shift Routine

```
      subroutine shift_data(psi,mynx,myny,
     &          shift_up, shift_down, shift_left, shift_right)

        include "mpif.h"
        integer mynx, myny
        real psi(mynx,myny)
        integer i,j,left, right, down, above, comm_2d,strided
        integer ierr, coords(2), stat(MPI_STATUS_SIZE)
        logical shift_up, shift_down, shift_left, shift_right
        common /n1/ comm_2d,coords,left,right,above,down
        common /n2/strided

        if (shift_up) then
        call mpi_send(psi(1,myny-1),mynx,MPI_REAL,above,0,
     &                               comm_2d,ierr)
        call mpi_recv(psi(1,1),mynx,MPI_REAL,down,0,
     &                        comm_2d, stat, ierr)
        endif

        if (shift_down) then
        call mpi_send(psi(1,2),mynx,MPI_REAL,down,1,
     &                             comm_2d, ierr)
        call mpi_recv(psi(1,myny),mynx,MPI_REAL,above,1,
     &                        comm_2d,stat, ierr)
        endif

        if (shift_right) then
        call mpi_send(psi(mynx-1,1), 1, strided, right,2,
     &                               comm_2d, ierr)
        call mpi_recv(psi(1,1), 1, strided, left, 2,
     &                          comm_2d, stat, ierr)
        endif
```

```
   if (shift_left) then
   call mpi_send(psi(2,1), 1, strided, left, 3,
&                                  comm_2d, ierr)
   call mpi_recv(psi(mynx,1), 1, strided, right,3,
&                            comm_2d, stat, ierr)
   endif

   return
   end
```

3.4 Performance Figures on Parallel Platforms

The platforms utilized included the CM5-E (with CMF), the Cray C90 and T90 (with -O3 auto-tasking), the Cray T3E (with HPF and MPI), the SGI Origin2000 (with pfa, HPF and MPI), the IBM SP2 (with HPF) and the Sun Global Works (with HPF). The MPI version of this code employed a 2-D tiling decomposition, and parallel runs were performed up to 512 processors on the T3E, up to 64 processors on the SGI Origin, and up to 64 on the Sun clusters.

Figure 3 shows the scalability of the code relative to multiples of 4 CPUs. The T3E with 64 processors achieved an 98 % scaling efficiency relative to 4 CPU's; we found that poor single PE performance held the overall speed down. For 512 processors, the scaling relative to 32 CPUs was 82 %. The Sun systems scaled well, with a 95 % efficiency, up to 32 CPUs, but then their scalability declined sharply. The SGI O2K with 32 processors showed only a 75 % scaling efficiency, but this improved to 84 % with 56 processors. With the O2K, the beneficial cache effects were pronounced with increasing CPUs.

Figure 4 depicts the total MFlops output of the various platforms as a function of CPUS and problem size. For the 2 GB problem size, the O2K efficiency with 64 CPUs is almost 100 % vs. 32 processors, but less than linear for a smaller number. The MFlop output for 56 processors was 3230. For the O2K, the MFlop output for the 4 GB problem size was slightly higher than for the 2 GB size with 16 CPUs; the 4 GB problem has not yet been tested for larger number of processors. We note that the 32 processor MFlop output for the T3E isonly a half of the O2K for 4 GB, but then it scales well up to 64 CPUs.

The auto-tasking versions were quite efficient even for small program sizes (17 Mb) and for small number of processors, with the SGI -pfa compiler option (with -O2 optimization) giving scalings of 1.9, 3.7, and 15.4 for 2, 4 and 16 CPU's, respectively, while the Cray T90 -O3 option (with FT90) gave scalings of 3.6 and 6.6 for 4 and 8 processors, respectively.

4 References

Bleck,R., S.Dean, M.O'Keefe and A.Sawdey, 1995: A Comparison of Data-Parallel and Message-Passing Versions of the Miami Isopycnic Coordinate Ocean Model. Parallel Computing 21, pp.1695-1720.

Blumberg,A.F. and G.L.Mellor, 1987: A Description of a Three-Dimensional Coastal Ocean Circulation Model. In 'Three-Dimensional Coastal Ocean Models', Coastal and Estuarine Sciences, AGU, Washington, D.C., pp.1-16.

Bryan,K., 1969: A numerical method for the study of the circulation of the world ocean. J. Comput. Phys. 4, p.347.

Bryan,K., 1979: Models of the world ocean circulation. Dynamics of Atmospheres and Oceans 3, p.327.

Buzbee,B.L.,F.W.Dorr,L.A.George and G.H.Golub, 1971: The direct solution of the discrete Poisson equation on irregular regions. SIAM J. Numer. Anal. 8, p.722.

Dukowicz,J.K and R.D.Smith, 1994: Implicit free-surface model for the Bryan- Cox-Semtner ocean model. J. Geophys. Res. 99, p.7991.

Dukowicz,J.K, R.D.Smith and R.C.Malone, 1993: A reformulation and implementation of the Bryan-Cox-Semtner ocean model on the connection machine. J. Atmos. Ocean. Technol. 10, p.195.

Kantha,L.H., 1995: Barotropic Tides in the Global Ocean from a Nonlinear Tidal Model Assimilating Altimetric Tides, Part I. Model Description and Results. J. Geophys. Res. 100, pp.25283-25308.

Ma,H., J.McCaffrey and S.Piacsek, 1998: A Parallel Implementation of a Spectral Element Ocean Model for Simulating Low-Latitude Circulation System. in Parallel Computational Fluid Dynamics, pp.641-648, ed. D.R.Emerson, A.Ecer, J.Periaux, N.Satofuka and P.Fox. Elsevier Press.

Madala,R.V. and S.A.Piacsek, 1977: A Model for Baroclinic Oceans. J. Comp. Phys. 23, p.167.

Piacsek,S.A. and R.Allard, 1994: Barotropic Coastal Currents in the Mediterranean. In Proceedings of the Second International Conference on Air-Sea Interaction and on Meteorology and Oceanography of the Coastal Zone. American Meteorological Society, Boston, MA, p.206.

Oberhuber,J.M. and K.Ketelsen, 1995: Parallelization of an OGCM on the CRAY T3D. Internal report, Deutsches Klimarechenzentrum GmbH, Model Support Group.

Piacsek,S.A. and A.J.Wallcraft, 1993: Initial Experiences with the Connection Machine at NRL. NRL Technical Note # 73-5089-03.

Schwarztrauber,P.N., 1977: The methods of cyclic reduction, Fourier analysis and cyclic reduction, and Fourier analysis - for the discrete solution of Poisson's equation on a rectangle. SIAM Rev. 19, p.490.

Smith,R.D., J.K.Dukowicz and R.C.Malone, 1992: Parallel ocean general circulation modeling. Physica D Amsterdam 60, p.38.

Wallcraft,A.J., 1991: The Navy Layered Ocean Model Users Guide. NOARL Report # 35, Dec. 1991, 21 pp. Stennis Space Center, MS 39529, USA

Wallcraft,A.J. and D.R.Moore, 1997: The NRL layered ocean model. Parallel Computing 23, pp. 2227-2242.

Webb,D.J., 1995: An ocean model code for array processor computers. Internal Document No.324, Institute of Oceanographic Sciences, Wormley, U.K.

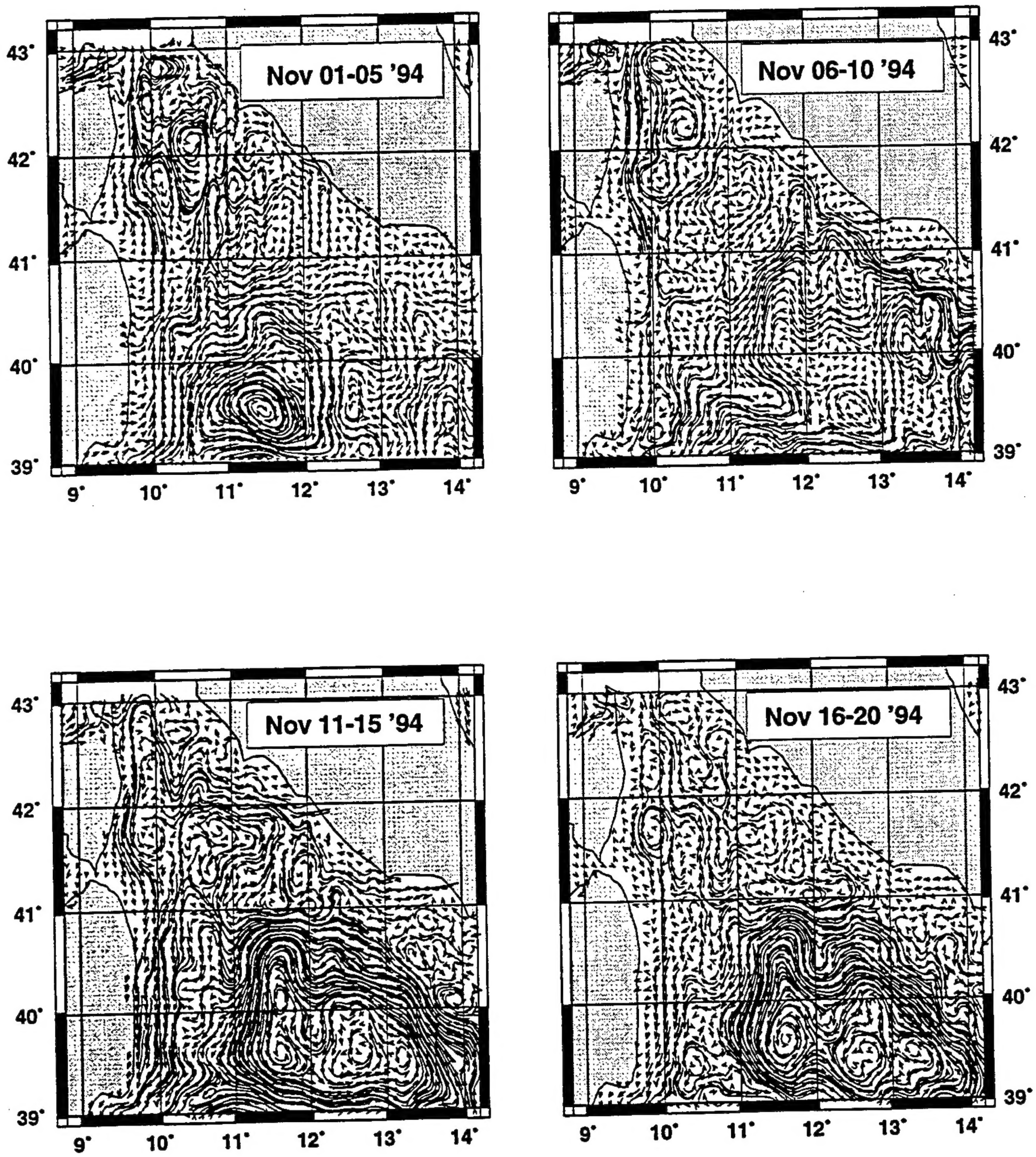

Figure 1: The barotropic circulation in the Tyrrhenian Sea, for the month November 1994.

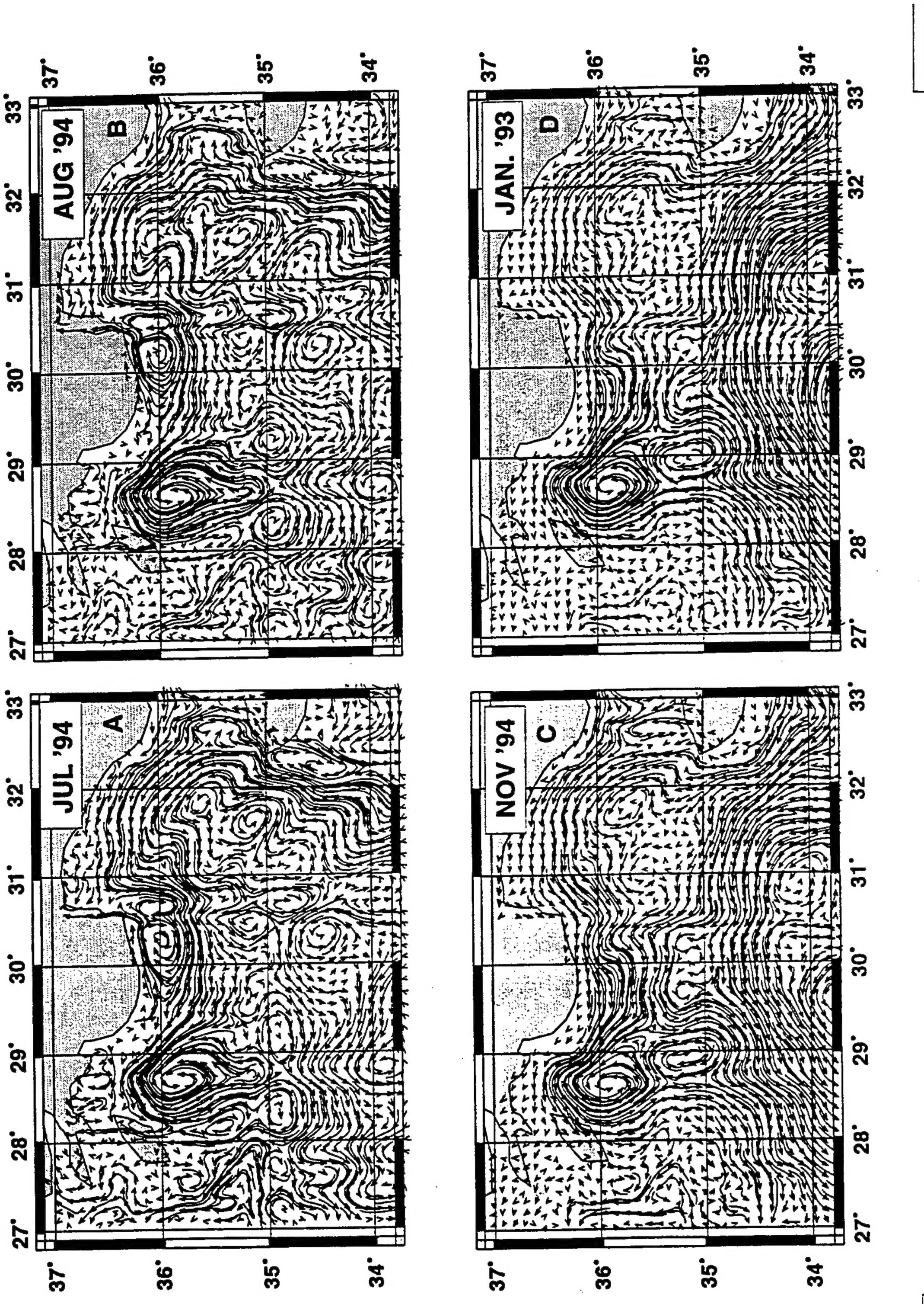

Figure 2: Seasonal evolution of the barotropic circulation in the NE quadrant of the Eastern Mediterranean, for 1994.

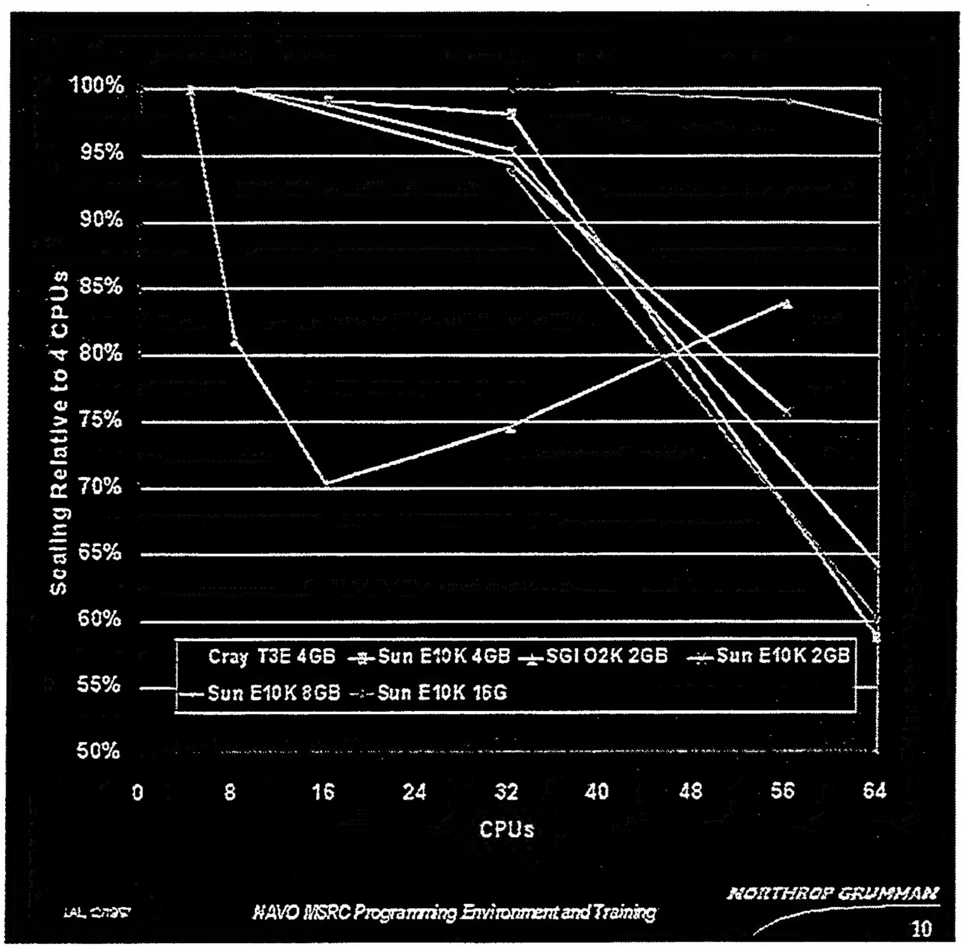

Figure 3: Scalability of the MPI version of the barotropic code relative to 4 CPUs

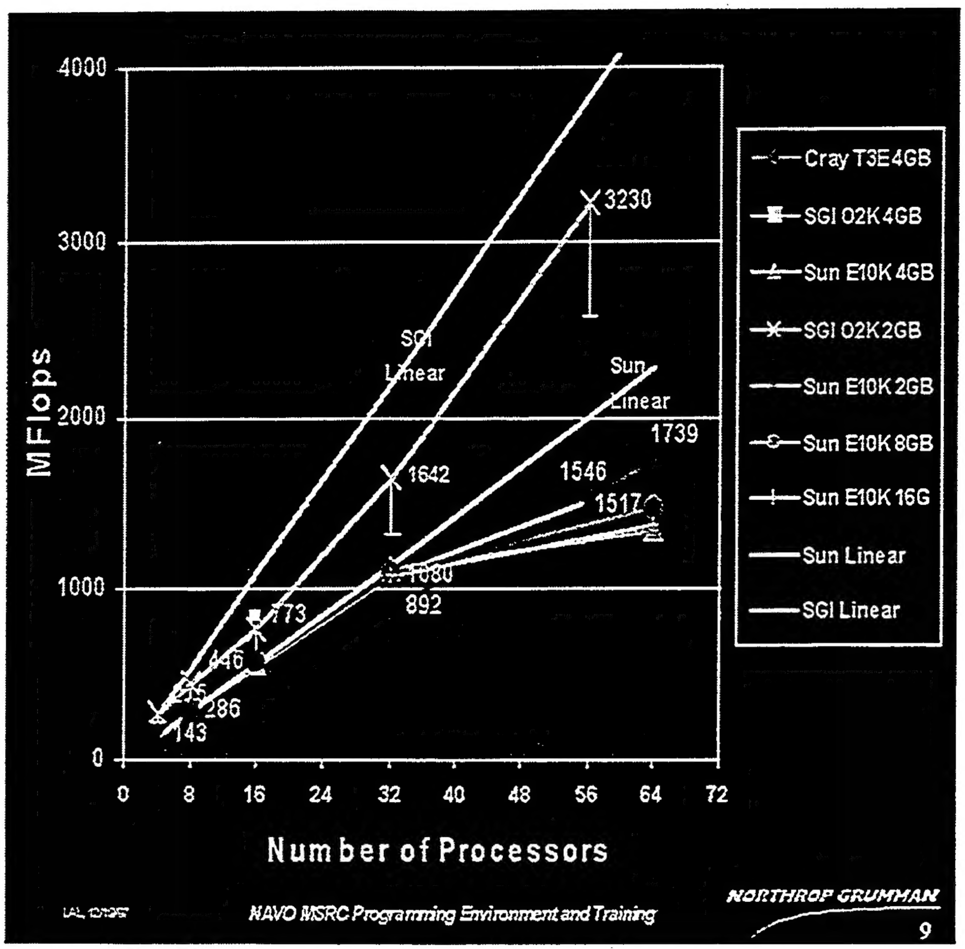

Figure 4: Scalability of the MPI version of the barotropic code in terms of MFlops